JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 891-905 (2017)
DOI: 10.6688/JISE.2017.33.4.2

Block-Level Message-Locked Encryption with Polynomial Commitment for IoT Data

KE HUANG, XIAO-SONG ZHANG AND XIAO-FEN WANG
Center for Cyber Security
University of Electronic Science and Technology of China
Gaoxin District, Chengdu, 611731 P.R. China
E-mail: kevinhuanguestc@163.com; johnsonzxs@uestc.edu.cn; wangxuedou@sina.com

The promise of smart city leads to store massive sensitive IoT data in cloud storage from various sources. Storage burden and security concerns are the most challenging issues. Message-Locked Encryption (MLE) and Proof of Storage (PoS) are useful tools to solve these problems. MLE encrypts data meanwhile enabling deduplication on them to save storage, and PoS checks data integrity in case of any data corruptions. However, trivial combination of PoS with MLE results in additional metadata which contradicts with the aim of deduplication. Therefore, how to integrate PoS with MLE for IoT data is an interesting research problem. To solve this problem, we propose a block-level message-locked encryption scheme with polynomial commitment for IoT data, called BL-MLE-PC. We introduce a unique set of metadata called Quadruple Tags (QTs) which can serve as: block identifiers, PoW tags, PoS tags and decryption keys. In addition, we use polynomial commitment to obtain fast and efficient data auditing. Our scheme can deduplicate under block-level for fine-grained saving. We prove that our scheme is secure under predefined security models. The analysis shows our scheme is efficient.

Keywords: smart city, message-locked encryption, polynomial commitment, deduplication, IoT data

1. INTRODUCTION

Smart city has been identified as an exemplar answer for many global issues (environment, healthcare, economy and governance), because this city is expected to run efficiently, sustainably and intelligently. The key enabler of smart city is the Internet of Things (IoT) in which all objects are connected and able to exchange data [1]. Originated from various sources (sensors, phones, *etc.*), IoT data provides valuable insights to improve urban performance [2]. For example, exploiting water data helps improve environment and prevent disaster [3]. Before use, IoT data needs to be transformed into proper format and stored in storage [4]. Cloud storage is a desirable platform to keep IoT data as it achieves economies of scale and supports convenient outsourcing services [5, 6]. However, the amount of IoT data is massive and it increases rapidly. Moreover, IoT data reflects sensitive information of cities, its leakages may cause economic and political loss [7]. Thus, an efficient and safe way to store IoT data in cloud is critical to the feasibility of smart city.

Many cloud providers adopt deduplication to relieve storage burden. Deduplication works by deleting repeated data and keeping only one copy in storage. Message-Locked

Received May 21, 2016; revised July 11, 2016; accepted August 30, 2017.
Communicated by Zhe Liu.
* This work is funded by National Natural Science Foundation of China under grant No. 61502086 and 61572-115; the Sichuan Provincial Major Frontier Issues (2016JY0007).

Encryption (MLE) [8] can encrypt file while enabling deduplication on them to save storage. An MLE scheme comprises five algorithms: $\mathcal{P}$, $\mathcal{K}$, $\mathcal{E}$, $\mathcal{D}$, $\mathcal{T}$. For example, suppose Alice has a file M to encrypt. Denote $P = \mathcal{P}(1^{\lambda})$ as public parameter and λ as security parameter. Alice first computes $K = \mathcal{K}(M, P)$ as encryption key. Then, she generates $C = \mathcal{E}(M, K, P)$ as ciphertext. Next, she computes $T = \mathcal{T}(C, P)$ as file tag. At last, C and T are outsourced to server while K is kept by Alice as decryption key. The server uses file tag T to detect replicas. Since identical file M results in same tag T, repeated files can be found. To guarantee secure deduplication, Proof of Ownership (PoW) [9] is adopted to verify file ownership between user and server. Any adversaries intend to pass this protocol without entire file will be detected with non-negligible possibility.

Proof of Storage (PoS) is often used to audit data integrity in cloud. There are PoS schemes like PDP [10-12] and POR [13-15] with different functionalities. But trivial combination of PoS with MLE leads to multiple set of metadata (PoS tags, PoW tags, decryption keys, *etc.*). This contradicts with the aim of deduplication. In this paper, we try to give the solution to this problem.

1.1 Related Work

We review four schemes [16-19] which integrate deduplication with PoS. We denote them all as PoWS, which indicates combination of PoW with PoS. Among them, Zheng and Xu proposed the first PoWS scheme, called POSD [16]. However, it is impractical due to its reliance on random key assumption. Then, Yuan *et al.* came up with a scheme called PCAD [17] where polynomial commitment is employed to achieve secure and constant PoS. Meanwhile, public auditing and batch auditing are also supported in PCAD. Next, Du *et al.* proposed PoOR [18] in which data can be recovered from corruption. These three schemes only support plaintext deduplication.

Recently, Chen *et al.* proposed a block-level message-locked encryption, called BL-MLE [19]. They achieved block-level deduplication under encryption as well as PoS. In their work, they computed hash of file as private key and used file tag as public key for auditing purpose. We point out that this method is insecure and explain why in section 5. Instead, we propose BL-MLE-PC for secure auditing while keeping advantages of BL-MLE.

1.2 Our Contribution

In this paper, we introduce a block-level message-locked encryption scheme with polynomial commitment (BL-MLE-PC) for IoT data. It ensures efficiency and security of IoT data stored in cloud for smart city. In our work, we introduce Quadruple Tags (QTs) which encapsulate all metadata (PoW tags, PoS tags, decryption keys and block identifiers) into single set. We utilize polynomial commitment to minimize PoS computation on user-side. We give security proof for pre-defined security models. The proof shows our scheme is secure against Chosen Distribution Attack (CDA) [8] and Duplicated Faking Attack (DFA) [8]. We compare BL-MLE-PC with other schemes to evaluate its performance. The analysis shows our scheme is efficient in use.

1.3 Paper Organization

The rest of this paper is organized as follows. Section 2 provides preliminary knowledge. Section 3 presents framework, threat model and security model. We show concrete construction of BL-MLE-PC in Section 4. Section 5 provides security analysis. Section 6 shows performance evaluation. Section 7 concludes this paper.

2. PRELIMINARY

In this section, we first give a brief introduction on symmetric bilinear map used in our scheme. Next, we give four complexity assumptions under which our security proofs are built on.

2.1 Symmetric Bilinear Map

Assuming there are two multiplicative groups $\boldsymbol{G}$ and $\boldsymbol{G}^T$. Order is prime number p and generator is g. A symmetric bilinear map is the map e: $\boldsymbol{G} \times \boldsymbol{G} \rightarrow \boldsymbol{G}^T$ such that $e(u^a, v^b) = e(u, v)^{ab}$ for all $u, v \in \boldsymbol{G}$ and $a, b \in \boldsymbol{Z}_P$. Particularly, e can be efficiently computed and $e(g, g) \neq 1$.

2.2 Complexity Assumption

Discrete Logarithm Assumption (DLA): Given $g^x \in G$ where $x \in \boldsymbol{Z}_P$, there is no polynomial time (PT) adversary $\mathcal{A}$ can compute x with non-negligible probability.

Computational Diffie-Hellman Assumption (CDH) [20]: Given $g^x, g^y \in G$ where $x, y \in \boldsymbol{Z}_P$, no PT adversary $\mathcal{A}$ can compute g^{xy} with non-negligible probability.

Static Diffie-Hellman Assumption (SDH) [21]: Choose $x \xleftarrow{R} \boldsymbol{Z}_p^*$. Given g, g^x and $h \in G$ where g is the generator of group G, there is no PT adversary $\mathcal{A}$ can compute h^x with non-negligible probability.

t-Strong Diffie-Hellman Assumption (t-SDH) [22]: Choose $x \xleftarrow{R} \boldsymbol{Z}_p^*$. Given a ($t$+1)-tuple $(g, g^x, \ldots, g^{x^t}) \in G^{t+1}$ where g is the generator of q-order cyclic group G, the probability $\Pr[\mathcal{A}dv(g, g^x, \ldots, g^{x^t}) = (c, g^{\frac{1}{x+c}})]$ for any PT adversary $\mathcal{A}$ is negligible for any value of $x \in \boldsymbol{Z}_q^*/-x$.

3. MODELING BL-MLE-PC

In this section, we give system framework, threat model and security model of BL-MLE-PC.

3.1 The System Framework

The framework of BL-MLE-PC is depicted in Fig. 1. Our framework consists of four parties: the cloud server (CS), the trust authority (TA), the data owner and the deduplication user. At the beginning, the TA helps CS create some system parameters and will

go off-line afterwards. Denote data owner as one who intends to upload a file which does not exist on server. The CS runs PoW protocol with data owner to check file ownership. Denote deduplication user as one who uploads a file which already exists on server. If he passes PoW protocol, CS will return file access to him directly instead of proceeding upload. After passing PoW protocol, the data owner and deduplication user can initiate PoS protocol with CS to check data integrity at any time.

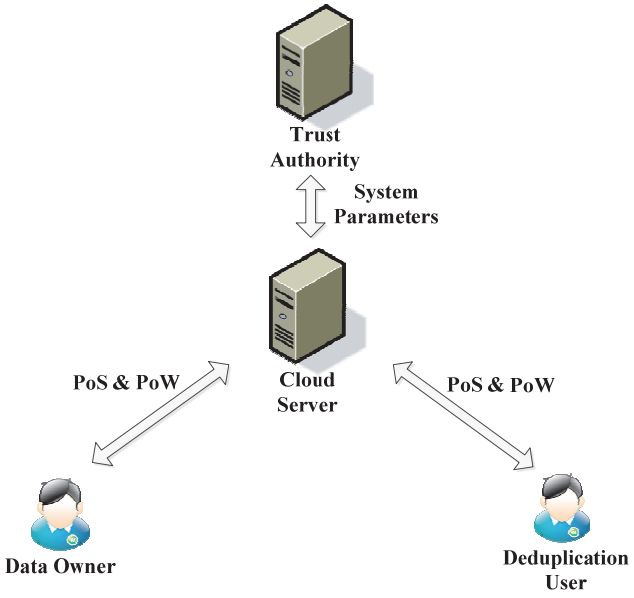


Fig. 1. BL-MLE-PC framework.

3.2 Threat Model

There are four possible threats against our scheme: PoS threats, privacy threats, tag consistency and PoW threats. We describe each one as below.

(A) PoS Threats

Three kinds of threats are related to PoS. First, attackers may corrupt data stored on cloud server. Second, cloud server may wreck data by misbehavior (software or hardware errors). Third, cloud server may delete rarely accessed data to save cost.

(B) Privacy Threats

Cloud server and malicious user are curious about the content of encrypted files, they intend to decrypt file and extract privacy from them. As files are encrypted deterministically, any users with partial information of file will successfully guess the complete file after polynomial trials. Such attack is known as Chosen Distribution Attack (CDA) [8]. As no MLE can satisfy any conventional semantic-style security, we therefore ask for the best security assuming files are unpredictable (with high min-entropy).

(C) Tag Consistency

Treats against tag consistency can be summarized as Duplicate Faking attack (DFA) [8]. In DFA, the adversary forges a file ciphertext C^* with the property which convinces server that C^* belongs to plaintext M. When valid file ciphertext C (corresponds to plaintext M) is uploaded by an honest user, the server discards C and only keeps C^*. Consequently, no one can retrieve the original plaintext M.

(D) PoW Threats

Similar to privacy threat, attackers with partial information of file may deploy CDA to pass the PoW protocol.

3.3 Security Model

A block-level message-locked encryption scheme with polynomial commitment (BL-MLE-PC) is secure if it meets the following four requirements: (1) PoS is secure; (2) privacy is secure against CDA; (3) tag consistency is secure against DFA; (4) PoW is secure against CDA. We capture security requirements by subsequent definitions and give proof in Section 5.

Definition 1: BL-MLE-PC scheme is PoS-secure if no adversary can forge to pass PoS verification with non-negligible advantage.

Definition 2: BL-MLE-PC scheme is privacy-secure if no polynomial time (PT) adversary $\mathcal{A}$ for any unpredictable block source S has non-negligible advantage in the PRV$-CDA-B game.

Definition 3: BL-MLE-PC is tag consistent if no PT adversary $\mathcal{A}$ has non-negligible advantage in the DFA game.

Definition 4: BL-MLE-PC scheme is PoW-secure if no PT adversary $\mathcal{A}$ has non-negligible advantage in the CDA game.

4. THE PROPOSED BL-MLE-PC

In this section, we give concrete construction of BL-MLE-PC and explain each algorithm explicitly. Next, we provide further explanation of our design. After that, we show algorithm correctness.

4.1 The Proposal

Setup Input a security parameter λ, output two groups $\boldsymbol{G}$, $\boldsymbol{G^T}$ and a bilinear map e: $\boldsymbol{G}\times \boldsymbol{G} \rightarrow \boldsymbol{G}^T$ (generator is g and order is prime p). Set integer s as number of sectors in each block. Define three hash functions H_a: $\{0, 1\} \rightarrow \boldsymbol{Z}_P$, H_b: $\{\boldsymbol{Z}_P\}^S \rightarrow \boldsymbol{G}$, H_c: $\boldsymbol{G} \rightarrow \{\boldsymbol{Z}_P\}^S$. Pick a random number $\alpha \xleftarrow{R} \boldsymbol{Z}_p^*$ and generate coefficients $\{g^{\alpha^j}\}_{1\le j\le s}$ for polynomial commitment [23]. Output system parameters:

$P = <p, g, \boldsymbol{G}, \boldsymbol{G}^{\boldsymbol{T}}, H_a, H_b, H_c, s, \{g^{\alpha^j}\}_{1\le j\le s}>$

KeyGen Input a file massage M and apply erasure code [24] to derive $\{M[i][j]\}_{1\le i\le n, 1\le j\le s}$. Compute master key $k_M = H_a(M)$. For each block $M[i]$, compute block key $k_i = H_b(M[i])$. Output master key k_M and block keys $\{k_i\}_{1\le i\le n}$.

Encrypt Input block plaintext $M[i]$ and block key k_i, output block ciphertext:

$C[i] = H_c(k_i) \oplus M[i]$.

Here, each block ciphertext $C[i]$ is generated by bit-wise XOR computation.

TagGen Input file $M = M[1]|| \ldots ||M[n]$, compute file tag $\sigma_0 = g^{k_M}$. For $1 \le i \le n$, compute each block tag σ_i:

$$\sigma_i = (k_i \cdot \prod_{j=1}^{s} g^{\alpha^j c[i][j]})^{k_M}.$$

For $1 \le i \le n$, compute each auxiliary information aux_i:

$aux_i = e(k_i, \sigma_0)$.

Output $\{\sigma_0, \{\sigma_i\}_{1\le i\le n}, \{aux_i\}_{1\le i\le n}\}$.

ConTest Input block ciphertext $C[i]$, block tag σ_i and auxiliary information aux_i, check whether:

$$e(\sigma_i, g) \stackrel{?}{=} e(\prod_{j=1}^{s} g^{\alpha^j c[i][j]}, \sigma_0) \cdot aux_i.$$

If holds, output 1; otherwise, output 0.

EqTest Input two block tags σ_i, σ_t' and corresponding file tags σ_0 and σ_0'. Check whether:

$$e(\sigma_i, \sigma_0') \stackrel{?}{=} e(\sigma_t', \sigma_0).$$

If holds, output 1; otherwise, output 0.

PoW-Challenge No input, the server randomly chooses c elements from $[1, n]$ to form a set $Q = \{(i, v_i)\}_{1\le i\le c}$. Output a challenge message $CM1 = \{(i, v_i)\}_{1\le i\le c}$.

PoW-Prove Input a challenge message $CM1$, output a proof message $Prf1 = \prod_{(i,v_i)\in Q} \sigma_i^{v_i}$.

PoW-Verify Input a proof message $Prf1$ and challenge message $CM1$, compute $Prf_T = \prod_{(i,v_i)\in Q} \sigma_i^{v_i}$. Check whether $Prf1 \stackrel{?}{=} Prf_T$. If holds, output 1; otherwise, output 0.

PoS-Challenge No input, pick k random elements from $[1, n]$ to form a set K and

choose a random number $r \xleftarrow{R} Z_p^*$. Output a challenge message $CM2 = \{K, r\}$.

PoS-Prove Input a challenge message $CM2$, compute $\{p_i = r^i \bmod p, i \in K\}$. Then, compute $y = f_{\vec{A}}(r)$ where polynomial coefficients vector $\vec{A} = \{0, 0, \sum_{i\in K} p_i C[i][1], \ldots, \sum_{i\in K} p_i C[i][s]\}$. As polynomials $f(x) \in Z[x]$ have algebraic property so that polynomials $f_{\vec{A}}(x) - f_{\vec{A}}(r)$ can be perfectly divided by $(x - r)$ using polynomial long division [23]. Parse the resulting quotient polynomial as $\vec{w} = \{w_1, \ldots, w_s\}$. So, we have $f_{\vec{w}}(x) = \frac{f_{\vec{A}}(x) - f_{\vec{A}}(r)}{x - r}$. Next, compute $\psi = \prod_{j=1}^{s} g^{\alpha^j w_j} = g^{f_{\vec{w}}(\alpha)}$. Denote $D = \{d_i\}_{i\in K} = \{\prod_{j=1}^{s} g^{\alpha^j C[i][j]}\}_{i\in K}$ and $E = \{\sigma_i\}_{i\in K}$. Compute $\sigma = \sum_{i\in K} \sigma_i^{p_i}$. Output a proof message $Prf2 = \{\sigma, \psi, y, E, D\}$.

PoS-Verify Input a proof message $Prf2$, compute $\eta = \sum_{i\in K} k_i^{p_i}$ and $v = g^{\alpha k_M}$. Check whether:

$$e(\eta, \sigma_0) \cdot e(\psi, v \cdot \sigma_0^{-r}) \stackrel{?}{=} e(\sigma, g) \cdot (\sigma_0^{-y}, g).$$

If holds, output 1; otherwise output 0.

KeyRet Input block ciphertext $C[i]$ and block tag σ_i, output block key k_i:

$$k_i = \sigma_i^{k_M^{-1}} \left(\prod_{j=1}^{s} g^{\alpha^j c[i][j]}\right)^{-1}.$$

Decrypt Input a block ciphertext $C[i]$ and block key k_i, output block plaintext $M[i]$:

$$M[i] = H_c(k_i) \oplus C[i].$$

4.2 Further Explanation

Although we require user to upload auxiliary information $\{aux_i\}_{1\le i\le n}$ to server, but it can be discarded soon after the server runs algorithm **ConTest** to check tag consistency. At last, the server keeps $\{C[i][j]_{1\le i\le n,\ 1\le j\le s}\}$, $\{\sigma_i\}_{0\le i\le n}\}$ in storage.

Since same block from distinct files results in different block tag, direct comparison of block tag for deduplication is impossible. Instead, we use algorithm **EqTest** to test equality. Given σ_i and σ'_t as block tag of $c[i]$ and $c'[t]$ respectively, corresponding file tags are σ_0 and σ'_0. The equation below holds when $c[i] = c'[t]$. This approach is also used in [8, 19, 25].

$$\begin{aligned} e(\sigma_i, \sigma'_0) &= \left((k_i \cdot \prod_{j=1}^{s} g^{\alpha^j c[i][j]})^{k_M}, g^{k'_M}\right) \\ &= \left((k_i \cdot \prod_{j=1}^{s} g^{\alpha^j c[i][j]}), g\right)^{k_M \cdot k'_M} \\ &= \left((k_i \cdot \prod_{j=1}^{s} g^{\alpha^j c[i][j]})^{k'_M}, g^{k_M}\right) \\ &= \left((k'_t \cdot \prod_{t=1}^{s} g^{\alpha^t c'[t][j]})^{k'_M}, g^{k_M}\right) \\ &= e(\sigma'_t, \sigma_0) \end{aligned}$$

4.3 Correctness

It is obvious that algorithm **ConTest**, **EqTest**, **Decrypt**, **PoS-Verify** and **PoW-Verify** satisfy correctness. Due to space limitation, we only give PoS correctness here.

$$
\begin{aligned}
& e(\eta,\sigma_0)\cdot e(\psi, v\cdot\sigma_0^{-r}) \\
&= e\left(\prod\nolimits_{i\in k} k_i^{p_i}, g^{k_M}\right)\cdot e\left(g^{f_{\tilde{w}}(\alpha)}, g^{\alpha k_M}\cdot g^{-rk_M}\right) \\
&= e\left(\prod\nolimits_{i\in k} k_i^{p_i}, g^{k_M}\right)\cdot e\left(g^{\frac{f_{\vec{A}}(\alpha)-f_{\vec{A}}(r)}{\alpha-r}}, g^{(\alpha-r)k_M}\right) \\
&= e\left(\prod\nolimits_{i\in k}\left[k_i\cdot g^{\alpha^j c[i][j]}\right]^{k_M p_i}, g\right)\cdot e\left(\sigma_0^{-y}, g\right) \\
&= e\left(\prod\nolimits_{i\in k}\sigma_i^{p_i}, g\right)\cdot e\left(\sigma_0^{-y}, g\right) \\
&= e(\sigma, g)\cdot e(\sigma_0^{-y}, g)
\end{aligned}
$$

5. SECURITY ANALYSIS

In this section, the security proof of BL-MLE-PC is shown.

5.1 PoS Security

We first explain why PoS extension of BL-MLE is insecure. In BL-MLE, master key $k_M = H_a(M)$ is taken as private key for PoS. However, this method can be viewed as "hash-as-a-proof", which is insecure according to [26]. The reason is: hash function H_a is publicly known, anyone with file M can derive k_M in a deterministic way. Moreover, cloud server can easily get k_M elsewhere (*e.g.* by colluding with other data owners). Thus, server can provide valid PoS proof without file.

To solve above problem, we introduce Quadruple Tags (QTs) for PoS in this paper. Inspired by [17], we ask to check: $e(\eta, \sigma_0) \cdot e(\psi, v \cdot \sigma_0^{-r}) \stackrel{?}{=} e(\sigma, g) \cdot e(\sigma_0^{-y}, g)$. The server needs to provide proof information $\{\sigma, \psi, y, E, D\}$ based on challenge. As each block ciphertext is enumerated in computation, it is hard to forge a proof without storing entire file. Proof is shown below.

Theorem 1: If there exists a PT adversary $\mathcal{A}$ that can forge a valid proof message $Prf2' \neq Prf2$ to pass PoS verification, then an algorithm $\mathcal{B}$ can use $\mathcal{A}$ to efficiently solve CDH problem, SDH problem or t-SDH problem.

The proof message in our scheme is $Prf2 = \{\sigma, \psi, y, E, D\}$. Compared with PCAD [17], $Prf2$ involves two more information (E and D). E denotes aggregation of challenged block tags and D denotes information of challenged blocks. Basically, E and D are used to recover block keys of challenged blocks $\{k_i\}_{i\in K}$.

The rest proof is clear based on [ThIII-5, 17]. Hence, we omit it here.

5.2 Privacy

Depending on the hardness of Computation Diffie-Hellman problem (CDH), we prove privacy security as below.

We start by reviewing some relevant knowledge. Notion of PRV-CDA is based on hedged PKE [27] which asks for indistinguishability of encryptions of two unpredictable messages [8]. Notion of PRV$-CDA is stronger in the sense that it asks that encryptions of unpredictable messages are indistinguishable from random strings. Particularly, Chen *et al.* modified notion of PRV$-CDA slightly to PRV$-CDA-B in their work [19]. The reason for modification is that BL-MLE generates block tags whereas MLE does not.

The proof is in two steps. First, we show PRV$-CDA-B attack on our BL-MLE-PC scheme can be converted to PRV$-CDA-B attack on BL-MLE. Then, we show BL-MLE is PRV$-CDA-B secure if CDH assumption holds.

Theorem 2: Suppose $\mathcal{A}$ is a PRV$-CDA-B adversary that has advantage $\epsilon(\lambda)$ against BL-MLE-PC scheme. $\mathcal{A}$ makes at most $q_a(\lambda)$, $q_b(\lambda)$ and $q_c(\lambda)$ queries to hash function H_a, H_b and H_c respectively. Then there is a PRV$-CDA-B adversary $\mathcal{B}$ that has advantage $\epsilon(\lambda)$ against BL-MLE scheme. Its running time is $O(time(\mathcal{A}))$.

***Proof*:** We briefly describe how to construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to gain advantage $\epsilon(\lambda)$ against BL-MLE scheme.

Setup: The adversary $\mathcal{A}$ initiates the game by sending information of an unpredictable block-source $\mathcal{S}$ to $\mathcal{B}$, $\mathcal{B}$ relays $\mathcal{S}$ to the challenger. On receiving $\mathcal{S}$, the challenger runs algorithm **Setup** in BL-MLE to generate system parameters $P = <p, g, \boldsymbol{G}, \boldsymbol{G}^T, H_1, H_2, H_3, s, u_1, \ldots, u_s>$ and sends to adversary $\mathcal{B}$. Here, hash function H_1, H_2 and H_3 in BL-MLE corresponds to H_a, H_b and H_c in our scheme. Set each $g^{a^j} = u_j$ for $(1 \le j \le s)$ and replace H_1, H_2, H_3 with H_a, H_b, H_c. Adversary $\mathcal{B}$ sends $P' = <p, g, \boldsymbol{G}, \boldsymbol{G}^T, H_a, H_b, H_c, s, g^a, \ldots, g^{a^s}>$ to $\mathcal{A}$. $\mathcal{B}$ answers hash queries from $\mathcal{A}$ by modeling hash functions: H_a, H_b, H_c as random oracles.

Challenge: The challenger randomly chooses $b = 0$ or $b = 1$. Let $n(\lambda)$ be the block numbers. For $b = 0$, the challenger runs the block-source $\mathcal{S}$ to output $(M_0, Z) \leftarrow M(\lambda)$. For $b = 1$, the challenger constructs M_1 uniformly at random from $\{0, 1\}^{|M_0|}$ and sets $M = M_b$. For each $i = 1$, $n(\lambda)$, the challenger runs algorithm **M-KeyGen** and **B-KeyGen** to compute master key k_M and block key k_i. Afterwards, the challenger runs algorithm **Encrypt, M-TagGen** and **B-TagGen** to generate each message ciphertext $C[i]$, block tag σ_i for $1 \le i \le n$ and file tag σ_0. Set $\sigma = \{\sigma_0, \sigma_1, \ldots, \sigma_{n(\lambda)}\}$. At last, the challenger sends auxiliary information Z, file ciphertext C and σ to the adversary $\mathcal{B}$. $\mathcal{B}$ relays (C, σ, Z) to $\mathcal{A}$.

Guess: On input (C, σ, Z), adversary $\mathcal{A}$ sends his guess b' to adversary $\mathcal{B}$. $\mathcal{B}$ outputs b' as his guess. The game succeeds if $b' = b$.

Theorem 3: If there is a PRV$-CDA-B adversary $\mathcal{A}$ with $\epsilon(\lambda)$ advantage against BL-MLE scheme, then there is an algorithm $\mathcal{B}$ can solve CDH problem with advantage $Adv^{\mathcal{B}}_{CDH}(\lambda)$ as below:

$$Adv_{CDH}^{\mathcal{B}}(\lambda) \geq \frac{2\epsilon(\lambda)}{q_{b,c}(\lambda)n(\lambda)} - \frac{n(\lambda)\cdot q_{b,c}(\lambda)}{2^{\mu(\lambda)}} - \frac{q_a(\lambda)}{2^{n(\lambda)\cdot\mu(\lambda)}}.$$

Proof of Theorem 3 is shown in [ThV-B, 19] and is hence omitted here. Thus, we reduce privacy security to CDH problem by Theorems 2 and 3.

5.3 Tag Consistency

We prove our scheme is tag consistent in two steps. First, we show DFA attack against BL-MLE-PC can be converted to DFA attack against BL-MLE. Next, we show BL-MLE scheme is tag consistent if Discrete Logarithm Assumption (DLA) holds.

Theorem 4: If there is a DFA adversary $\mathcal{A}$ with advantage $\epsilon(\lambda)$ against BL-MLE-PC, then there is a DFA adversary $\mathcal{B}$ that has advantage $\epsilon(\lambda)$ against BL-MLE.

***Proof*:** We briefly describe how to construct an adversary $\mathcal{B}$ with advantage $\epsilon(\lambda)$ against BL-MLE by using $\mathcal{A}$.

Setup: The challenger generates and sends $\mathcal{B}$ the system parameters $P = <p, g, \boldsymbol{G}, \boldsymbol{G}^T, H_1, H_2, H_3, s, u_1, \ldots, u_s>$. Adversary $\mathcal{B}$ generates and sends system parameters $P' = <p, g, \boldsymbol{G}, \boldsymbol{G}^T, H_a, H_b, H_c, s, g^a, \ldots, g^{a^l}>$ to $\mathcal{A}$.

Guess: $\mathcal{A}$ outputs $<M^*, i, C^*, \sigma^*>$ to $\mathcal{B}$, $\mathcal{B}$ relays it to the challenger. If $ConTest(C^*, \sigma^*) = 1$, $M^*[i] \neq Dec(KeyGen(M^*[i], C^*))$ and $EqTest(TagGen(M^*[i], \sigma^*)) \rightarrow 1$, output 1; otherwise output 0. The game succeeds if the output is 1.

Theorem 5: If a DFA adversary $\mathcal{A}$ can break BL-MLE scheme with advantage $\epsilon(\lambda)$, then there is an algorithm $\mathcal{B}$ can break the DLA with advantage more than $\frac{\epsilon(\lambda)}{2}$.

Theorem 5 is clear based on [ThV-C, 19]. Thus, we reduce security of tag consistency to DLA problem by Theorems 4 and 5.

5.4 PoW Security

We show PoW-security as below.

Theorem 6: A CDA adversary $\mathcal{A}$'s advantage in a CDA game is,

$$\epsilon(\lambda) \leq \frac{1}{2^{\mu(\lambda)}} \cdot \left(1 - \left(\frac{1+t(\lambda)-q(\lambda)}{1+n(\lambda)-q(\lambda)}\right)^{q(\lambda)} + \left(\frac{t(\lambda)}{n(\lambda)}\right)^{q(\lambda)}\right).$$

Concrete proof is shown in [ThV-D, 19] and is hence omitted here.

6. PERFORMANCE ANALYSIS

In this section, we evaluate scheme performance by comparing computation complexity and conducting simulation.

For comparison, we combine block-level message-locked encryption [19] with Yuan's PCAD [17] and Du's PoOR [18] for deduplication. Furthermore, we adopt Merkle Hash Tree (MHT) [9] as PoW protocol to authenticate file ownership. Denote the derived schemes as E-PCAD and E-PoOR respectively. To notice, the derived schemes produce four sets of metadata: block identifiers, PoS tags, decryption keys and PoW tags. In these derived schemes, user encrypts decryption keys with master key and uploads them to server. The maser key is kept secretly by user.

6.1 Complexity

We list computational complexity of BL-MLE-PC, E-PCAD and E-PoOR in different stages in Table 1. For ease of read, the operation symbol denotes meaning: C_{EXP}: exponential operation; C_{MUL}: multiplicative operation; $C_{Pairing}$: bilinear pairing operation; C_{MHT}: constructing a Merkle Hash Tree; C_{HASH}: hash operation; C_{XOR}: exclusive OR operation; C_{PRF}: pseudo-random function operation.

Table 1. Comparison of computation complexity.

	BL-MLE-PC	E-PCAD	E-PoOR
TagGen	$(ns - n + 1)C_{EXP} + nsC_{MUL} + nC_{Pairing}$	$(ns - n + 1)C_{EXP} + nsC_{MUL} + C_{MHT}$	$(s + 1)C_{PRF} + sC_{MUL}$
Encrypt	$nC_{HASH} + nC_{XOR}$	$nC_{HASH} + nC_{XOR}$	$nC_{HASH} + nC_{XOR}$
PoS-Prove	$(k + 2s - 2)C_{MUL} + (2s + k)C_{EXP}$	$(k + s - 1)C_{MUL} + (s + k)C_{EXP}$	$2sC_{PRF} + (s + k)C_{MUL}$
PoS-Verify	$(3k + 3)C_{EXP} + (2k + 2)C_{MUL} + 4C_{Paring}$	$C_{EXP} + 3C_{MUL} + 4C_{Paring}$	$(4s + 2k)C_{PRF} + (s + k)C_{MUL}$
EqTest	$(2 + 2n)C_{Pairing}$	nC_{HASH}	C_{HASH}
PoW-Prove	$qC_{EXP} + (q - 1)C_{MUL}$	C_{MHT}	C_{MHT}

k is the number of challenged blocks.
n is the number of blocks in each file and s is the number of sectors in each block.

6.2 Simulation

Next, we conduct a simulation on JAVA to test the performance of BL-MLE-PC. The platform is a 2.3GHz Intel i5-3210M CPU 4GB RAM laptop with 32 bits Windows 7 SP1 Operating System. Our experiment is implemented under 5Mbps bandwidth network environment. We set the security parameter $\lambda = 160$. We limit block size to 4KB and set number of challenged blocks to $k = 480$ which guarantees 99.999% successful rate. We range file size from 4 MB to 512 MB. Same setting is used in [28, 29]. The processing time of our scheme and direct upload is shown in Fig. 2.

In Fig. 2, the spending time of both cases increases with file size. Obviously, our scheme is more efficient than direct upload. The saving is more evident when file is large.

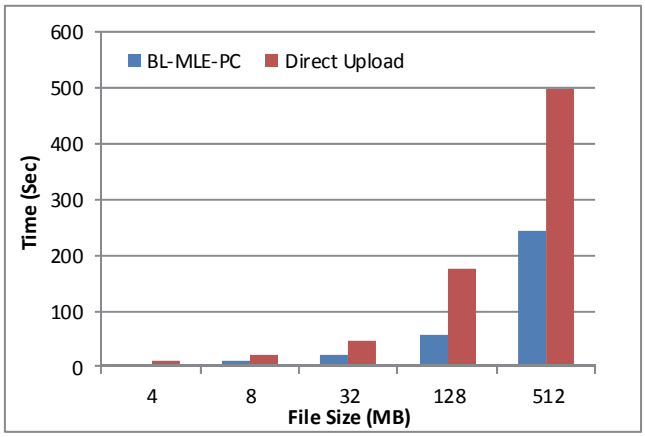


Fig. 2. Comparison of upload time.

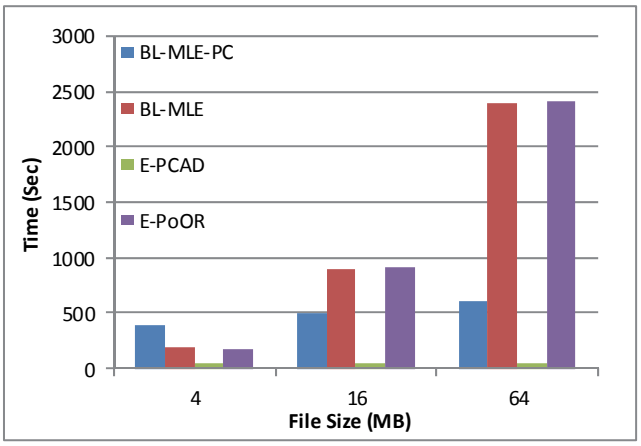


Fig. 3. Comparison of PoS time.

To test PoS performance, we vary file size from 4MB to 64MB. The number of challenged blocks k is fixed to 480. The experiment results are shown in Fig. 3.

In Fig. 3, the computation time of E-PCAD is constant due to the use of polynomial commitment. Meanwhile, the time spent by both BL-MLE and E-PoOR increase rapidly with file size. Despite less efficient than E-PCAD, our scheme still outperforms two others.

7. CONCLUSIONS

In this paper, we propose a block-level message-locked encryption scheme with polynomial commitment for IoT data. Our scheme provides efficient and secure management for IoT data stored in cloud storage. Above all, we introduce Quadruple Tags (QTs) to serve as block identifiers, decryption keys, PoW tags and PoS tags. The use of QTs guarantees that PoS will be performed in a secure way. We apply polynomial com-

mitment to QTs to reduce PoS computation. We build security model and our proof shows our scheme is secure against CDA and DFA. The performance analysis confirms that our scheme is practical in use.

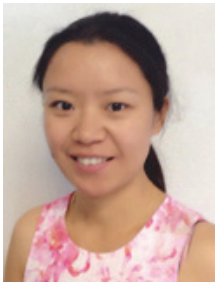
REFERENCES

1. S. Li, L. D. Xu, and S. Zhao, "The internet of things: A survey," *Information Systems Frontiers*, Vol. 17, 2015, pp. 243-259.
2. I. A. T. Hashem, V. Chang, N. B. Anuar, K. Adewole, I. Yaqoob, A. Gani, E. Ahmed, and H. Chiroma, "The role of big data in smart city," *International Journal of Information Management*, Vol. 36, 2016, pp. 748-758.
3. J. Shen, H. Tan, J. Wang, J. Wang, and S. Lee, "A novel routing protocol providing good transmission reliability in underwater sensor networks," *Journal of Internet Technology*, Vol. 16, 2015, pp. 171-178.
4. J. D. Bokefode, A. S. Bhise, P. A. Satarkar, and D. G. Modani, "Developing a secure cloud storage system for storing IoT data by applying role based encryption," *Procedia Computer Science*, Vol. 89, 2016, pp. 43-50.
5. Z. Fu, K. Ren, J. Shu, X. Sun and F. Huang, "Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, 2015, pp. 2546-2559.
6. Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Towards efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, 2016, p. 1.
7. Cloud Security Alliance, "Top threats to cloud computing," 2010, http://www.cloudsecurityalliance.org.
8. M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2013, pp. 296-312.
9. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of ACM Conference on Computer and Communications Security*, 2011, pp. 491-500.
10. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of ACM Conference on Computer and Communications Security*, Vol. 14, 2007, pp. 598-609.
11. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, 2010, pp. 847-859.
12. Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, Vol. 16, 2015, pp. 317-323.
13. A. Juels and B. S. Kaliski, "PORs: Proofs of retrievability for large files," in *Proceedings of ACM Conference on Computer and Communications Security*, 2007, pp. 584-597.

14. J. Yuan and S. Yu, "Proofs of retrievability with public verifiability and constant communication cost in cloud," *International Workshop on Security in Cloud Computing*, 2013, pp. 19-26.
15. H. Shacham and B. Waters, "Compact proofs of retrievability," *Journal of Cryptology*, Vol. 26, 2013, pp. 442-483.
16. Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in *Proceedings of ACM Conference on Data and Application Security and Privacy*, 2011, pp. 1-12.
17. J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with de duplication," in *Proceedings of IEEE Conference on Communications and Network Security*, 2013, pp. 145-153.
18. R. Du, L. Deng, J. Chen, K. He, and M. Zheng, "Proofs of ownership and retrievability in cloud storage," in *Proceedings of IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2014, pp. 328-335.
19. R. Chen, Y. Mu, G. Yang, and F. Guo, "BL-MLE: Block-level message-locked encryption for secure large file deduplication," *IEEE Transactions on Information Forensics and Security*, Vol. 10, 2015, pp. 2643-2652.
20. W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. 22, 1976, pp. 644-654.
21. D. R. L. Brown and R. P. Gallant, "The static Diffie-Hellman problem," *Cryptology ePrint Archive*, http://eprint.iacr.org/, 2004.
22. D. Boneh and X. Boyen, "Short signatures without random oracles," *Lecture Notes in Computer Science*, Vol. 3352, 2015, pp. 134-148.
23. A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *Proceedings of International Conference on Theory and Application of Cryptology and Information Security*, Vol. 6477, 2010, pp. 177-194.
24. I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for industrial and Applied Mathematics*, Vol. 8, 1960, pp. 300-304.
25. G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in *Proceedings of International Conference on Topics in Cryptology*, Vol. 5985, 2010, pp. 119-131.
26. J. Xu and J. Zhou, "Leakage resilient proofs of ownership in cloud storage, revisited," *Applied Cryptography and Network Security*, 2014, pp. 97-115.
27. M. Bellare, Z. Brakerski, M. Noar, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek, "Hedged public-key encryption: How to protect against bad randomness," in *Proceedings of International Conference on Theory and Application of Cryptology and Information Security: Advances in Cryptology*, Vol. 5912, 2009, pp. 232-249.
28. Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, 2016, pp. 1-12.
29. Z. Liu, H. Seo, J. Großschädl, and H. Kim, "Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes," *IEEE Transactions on Information Forensics and Security*, Vol. 11, 2015, pp. 1385-1397.

Ke Huang (黄可) received the M.S. degree from University of Electronic Science and Technology, China. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include data security and cryptography.

Xiao-Song Zhang (張小松) received the Ph.D. degree from University of Electronic Science and Technology of China. He is currently a Professor at University of Electronic Science and Technology of China. His research interests are software security, network security and data security.



Xiao-Fen Wang (汪小芬) received the Ph.D. degree from Xidian University in 2009. She is currently an Associate Professor at University of Electronic Science and Technology of China. She was a Visiting Research Fellow at University of Wollongong. Her research interests are cryptography and information security.